

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Modelling methods based on discrete algebraic systems

Hiroyuki Goto

*Nagaoka University of Technology  
Japan*

## 1. Introduction

A typical and significant feature of discrete event systems is that the behaviour is non-continuous; that is to say, events occur at discrete time instants and the values of internal states change non-continuously. A simple and well-known example of a discrete event system is a traffic signal network. Each signal has a current status of three possible states: green, yellow and red. Moreover, the status changes by predetermined time intervals, which are usually determined by inspecting past traffic conditions relevant to the site. A more complicated example is an air traffic control system at an airport. Clearances for take-off and landing issued by controllers can be understood as a sort of signal. However, the controller must take into account no-concurrency issues of the runway and the necessary time intervals for take-off or landing, as well as scheduled times. Thus, this system is more complex than the previous.

If we model and analyse such discrete event systems using the conventional formalism, we often have to incur specific constraints on the internal variables and parameters. For example, there are often cases whereby the explanatory variables have only Boolean (0/1) logical or integer values. This tends to make the formulation more complex and more difficult to solve. In view of this, several specific methods suited for discrete event systems, automaton (Kelarev, 2003) and Petri net (Girault & Valk, 2002) for instance, have been developed. These are modelling tools for simply representing the target systems, and are beneficial for analysing the behaviour of these; for example, so that critical sections such as so-called dead-lock or infinite-loops can be detected. The essence of these methods, however, is a kind of symbolisation, rather than formalisation. Thus, they are not suitable for taking into account varying parameters or structures.

Now let us go back to the essence of discrete event systems. What is the obstacle in using the conventional formalism? A primary point would be its non-linearity. In the above case of air traffic control, before clearance for take-off can be given to a pilot of an aircraft, the controller must check whether the runway is available, that no other aircraft is on or about to cross the same runway, and moreover is not in a take-off or landing phase. Several constraints of these are non-linear, but the non-linearity is weak. For instance, the status of whether multiple conditions are satisfied simultaneously is equivalent to the result of an 'and' operation. In terms of a time axis, clearance is given after the 'maximum' time of

which all necessary conditions are satisfied. Moreover, the phrase 'about to' can be interpreted as the result of considering a margin time, equivalent to time offset. As the above issues imply, several classes of discrete event systems may be formulated by combinations of simple non-linear functions. Accordingly, if we use algebraic systems suited for representing logical operations, 'and', 'or', 'max' and 'min' for instance, constraints may be formulated simply.

A most famous algebraic system would be the Boolean algebra (Harrison, 2009), which is popular in the field of electrical engineering and plays an essential role in designing logical circuits. In the Boolean algebra, the 'or' and 'and' operations are defined as logical addition and multiplication, respectively. Under these definitions, the essential properties in conventional algebra such as the laws of commutativity, associativity and distributivity are invoked in this algebraic system. With the help of this structure, several types of basic circuits can be modelled simply.

Another well-known structure is referred to as the Dioid algebraic system (Baccelli et al., 1992). This system requires defining two operators for addition and multiplication that satisfy the above laws. If we can determine a set of operators by which the constraints of the target system can be represented, the behaviour of the system would be formulated by simple equations. For instance, the max-plus algebra (Heidergott et al., 2006), a subclass of Dioid algebra, defines the 'max' and '+' operations as addition and multiplication, respectively. This algebraic system is suited for describing synchronisation of multiple events and time margins. This algebra is also referred to as the schedule algebra, and plays an essential role in this chapter. As this name implies, the max-plus algebra can be beneficially used in solving several classes of scheduling problems.

Let us now take a glance at the approach based on the max-plus algebra. In representing the behaviour of a target system, a set of linear equations is used. A simple and typical form is:  $\mathbf{x}(k) = \mathbf{A} \otimes \mathbf{x}(k-1) \oplus \mathbf{B} \otimes \mathbf{u}(k)$ ,  $\mathbf{y}(k) = \mathbf{C} \otimes \mathbf{x}(k)$ , where  $\oplus$  and  $\otimes$  represent the operators for addition and multiplication in the max-plus algebra, respectively. The reader familiar with control theory may already have noticed that the form is similar to the state-space representation in modern control theory. Hence, this approach is also compatible with concepts in control theory, and several research accomplishments in control theory have been applied to this field, typical research reports of which will be referred later. In addition to these, the max-plus algebraic system itself has a number of interesting features because of its specific definition. Thus, there is also a number of reports devoted to these pure mathematical aspects. Typical examples include the existence of solutions of simultaneous or polynomial equations, and eigenvalue problems.

As these issues indicate, for modelling and analysis methods for a class of discrete event systems, much attention has been paid to the approach based on the max-plus algebra. It seems though that there is less concerted research effort on extending the range of application and improving its practicability. For instance, the above state space representation has been sufficiently generalised and well-studied in past research. However, there is still little research on how to formulate systematically the behaviour of practical systems, which would be paramount when needed in applications to complex systems.

In view of this, we aim to improve the practicability of the state-space representation in Dioid and max-plus algebras. The basic concept and framework are explained in the subsequent section. Several recent developments are then introduced in the latter sections.

## 2. Preliminaries

This section first clarifies the research scope using simple illustrative examples, and then confirms the necessity of Dioid and max-plus algebras.

### 2.1 Simple example

The primary concern of this chapter is to provide a systematic framework for deriving the state-space representation for practical systems. Target applications include scheduling problems for a class of manufacturing systems.

Let us now consider the behaviour of a simple manufacturing system depicted in Fig. 1. The system has two external inputs, three facilities and one external output. Facility 1 receives raw material from input 1, processes it, and sends the resulting part to facility 3. The behaviour of facility 2 is the same as facility 1. Facility 3 receives the processed parts from facilities 1 and 2, processes them, and sends the resulting output to the external output. Assuming that this process is carried out repeatedly, let us derive the earliest process start times.

Facilities 1 and 2 can start processing after the processing of the previous part is completed and the required resource materials are fed. Moreover, facility 3 can start processing after the processing of the previous part is completed and the processed parts are received from facilities 1 and 2. For the  $k$ -th part, let us denote the earliest process start and processing time in facility  $i$  by  $x_i(k)$  and  $d_i$ , respectively. Moreover, we denote the material feeding times from external input  $i$  by  $u_i(k)$  and the earliest output time to the external output by  $y(k)$ . Then, the earliest process start and output times can be expressed in the following manner.

$$x_1(k) = \max \{x_1(k-1) + d_1, u_1(k)\}, \quad x_2(k) = \max \{x_2(k-1) + d_2, u_2(k)\} \quad (1)$$

$$x_3(k) = \max \{x_3(k-1) + d_3, x_1(k) + d_1, x_2(k) + d_2\} \quad (2)$$

$$y(k) = x_3(k) + d_3 \quad (3)$$

As is easily seen, all calculations consist of only two types of operations, max and +. Since the max operation is non-linear, the above equations are also non-linear in nature. However, if we use a specific discrete algebraic system, such types of equations can be represented by a set of linear equations. This can be accomplished by using Dioid algebra.

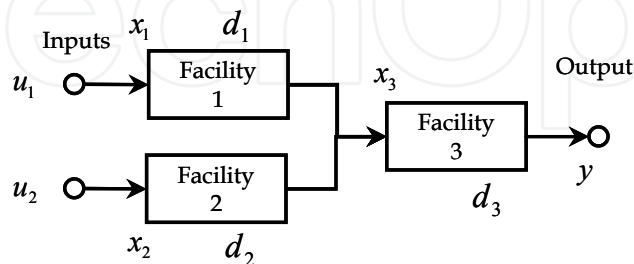


Fig. 1. A system with two inputs, one output and three facilities

### 2.2 Dioid and max-plus algebras

The basic concepts underlying Dioid and max-plus algebras are explained. Dioid algebra is defined in the field  $\mathbb{D}$  and endowed with a set  $\{\oplus, \otimes, \varepsilon, e\}$  consisting of two operators for

addition and multiplication, and two unit elements, respectively. For arbitrary elements  $x, y, z \in D$ , the following axioms, all of which are well-known in conventional algebra, are enforced:

$$\text{Commutative law: } x \oplus y = y \oplus x \quad (4)$$

$$\text{Associative law: } (x \oplus y) \oplus z = x \oplus (y \oplus z), (x \otimes y) \otimes z = x \otimes (y \otimes z) \quad (5)$$

$$\text{Distributive law: } (x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z), z \otimes (x \oplus y) = (z \otimes x) \oplus (z \otimes y) \quad (6)$$

For the unit elements  $\varepsilon$  and  $e$ , referred to as the zero and identity elements, we enforce the following properties:

$$x \oplus \varepsilon = x, x \otimes \varepsilon = \varepsilon \otimes x = \varepsilon, x \otimes e = e \otimes x = x, x \oplus x = x \quad (7)$$

We observe that only the last property is different from that in conventional algebra, and gives Dioid algebra its distinguishing and remarkable feature. Note that the Dioid is a collection of algebraic systems, and does not assume more specific operation rules.

As a subclass of Dioid algebra, max-plus algebra is endowed with a set  $\{\oplus, \otimes, \varepsilon, e\} \equiv \{\max, +, -\infty, 0\}$  defined in  $D \equiv R_{\max} = R \cup \{-\infty\}$ , where  $R$  represents the real field. As we can easily confirm, this set satisfies the above axioms (4)–(7), as follows:

$$\begin{aligned} x \oplus y &= \max(x, y) = y \oplus x \\ (x \oplus y) \oplus z &= \max(x, y, z) = x \oplus (y \oplus z), (x \otimes y) \otimes z = x + y + z = x \otimes (y \otimes z) \\ (x \oplus y) \otimes z &= \max(x, y) + z = \max(x + z, y + z) = (x \otimes z) \oplus (y \otimes z) \\ x \otimes (y \oplus z) &= x + \max(y, z) = \max(x + y, x + z) = (x \otimes y) \oplus (x \otimes z) \\ \max(x, -\infty) &= x, x + (-\infty) = (-\infty) + x = -\infty, x + 0 = 0 + x = x, \max(x, x) = x \end{aligned}$$

Max-plus algebraic system is a subclass of Dioid algebra, but it is not unique. For example, the following sets also satisfy the axioms (4)–(7).

$$\begin{aligned} \{\oplus, \otimes, e, \varepsilon\} &\equiv \{\min, +, +\infty, 0\} \text{ defined in } D \equiv R \cup \{+\infty\} \\ \{\oplus, \otimes, e, \varepsilon\} &\equiv \{\max, \times, +0, 1\} \text{ defined in } D \equiv R^+ \text{ (positive real field)} \\ \{\oplus, \otimes, e, \varepsilon\} &\equiv \{\max, \min, -\infty, +\infty\} \text{ defined in } D \equiv R \cup \{-\infty\} \cup \{+\infty\} \end{aligned}$$

We leave as an exercise for the reader to confirm that these sets also satisfy the axioms of the Dioid algebra.

Moreover, we adopt the notational rules for addition, multiplication and exponent in conventional algebra to this algebraic system. That is, we simply denote:

$$\bigoplus_{k=1}^l x_k \equiv x_1 \oplus x_2 \oplus \cdots \oplus x_l, xy \equiv x \otimes y, x^l \equiv \underbrace{x \otimes x \otimes \cdots \otimes x}_l$$

when no confusion is likely to arise.

Next, let us extend the max-plus algebraic system for scalars to matrices. For  $X \in \mathbf{R}_{\max}^{m \times n}$ ,  $Y \in \mathbf{R}_{\max}^{m \times n}$ ,  $V \in \mathbf{R}_{\max}^{n \times l}$ , we define the operational rules for addition and multiplication and unit elements in the following manner.

$$[X \oplus Y]_{ij} = \max([X]_{ij}, [Y]_{ij}), [X \otimes V]_{ij} = \bigoplus_{k=1}^l ([X]_{ik} \otimes [V]_{kj}) = \max_{k=1, \dots, l} ([X]_{ik} + [V]_{kj})$$

$\varepsilon$  : all elements are  $\varepsilon$

$e$  : only diagonal elements are  $e$  and all off-diagonal elements are  $\varepsilon$

Under these definitions, for arbitrary matrices  $X \in \mathbf{R}_{\max}^{m \times n}$ ,  $Y \in \mathbf{R}_{\max}^{m \times n}$ ,  $Z \in \mathbf{R}_{\max}^{m \times n}$ ,  $V \in \mathbf{R}_{\max}^{n \times l}$  and  $W \in \mathbf{R}_{\max}^{l \times m}$ , the following properties, essentially correspond to (4)–(7), hold true.

$$\begin{aligned} X \oplus Y &= Y \oplus X, (X \oplus Y) \oplus Z = X \oplus (Y \oplus Z), (X \otimes V) \otimes W = X \otimes (V \otimes W) \\ (X \oplus Y) \otimes V &= (X \otimes V) \oplus (Y \otimes V), W \otimes (X \oplus Y) = (W \otimes X) \oplus (W \otimes Y) \\ X \otimes \varepsilon &= X, X \otimes e = \varepsilon \otimes X = \varepsilon, X \otimes e = e \otimes X = X, X \oplus X = X \end{aligned}$$

We should note here that care is required with respect to the second and third relationships in (7). The sizes of the unit matrices  $\varepsilon$  and  $e$  must be adjusted in advance so that multiplication can be defined.

### 2.3 State space representation

We now simplify (1)–(3) using max-plus algebra. By replacing the max and + operations with  $\oplus$  and  $\otimes$ , respectively, the equations can be expressed as:

$$x_1(k) = x_1(k-1) \otimes d_1 \oplus u_1(k), \quad x_2(k) = x_2(k-1) \otimes d_2 \oplus u_2(k) \quad (8)$$

$$x_3(k) = x_3(k-1) \otimes d_3 \oplus x_1(k) \otimes d_1 \oplus x_2(k) \otimes d_2 \quad (9)$$

$$y(k) = x_3(k) \otimes d_3 \quad (10)$$

By substituting (8) into (9), we obtain:

$$x_3(k) = x_1(k-1) \otimes d_1^2 \oplus x_2(k-1) \otimes d_2^2 \oplus x_3(k-1) \otimes d_3 \oplus u_1(k) \otimes d_1 \oplus u_2(k) \otimes d_2 \quad (11)$$

We now notice that  $x_i(k)$  and  $y(k)$  are represented by linear functions of  $x_i(k-1)$  and  $u_i(k)$  in a max-plus algebraic system, and it seems that these can be simply expressed if we use a matrix representation. In fact, (8), (10), and (11) are summarised as follows.

$$\mathbf{x}(k) = \mathbf{A} \otimes \mathbf{x}(k-1) \oplus \mathbf{B} \otimes \mathbf{u}(k) \quad (12)$$

$$\mathbf{y}(k) = \mathbf{C} \otimes \mathbf{x}(k) \quad (13)$$



where:

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}, \mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}, \mathbf{y}(k) = [y(k)], \mathbf{A} = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon \\ d_1^2 & d_2^2 & d_3 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ d_1 & d_2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \varepsilon \\ \varepsilon \\ d_3 \end{bmatrix}^T$$

Equations (12) and (13) are referred to as the state and output equations, respectively. Moreover, the set of these equations is called the state-space representation. Variables  $\mathbf{x}(k)$ ,  $\mathbf{u}(k)$  and  $\mathbf{y}(k)$  are referred to as the state, input and output variables, respectively. Matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are referred to as the system, input and output matrices, respectively. A system whose behaviour can be described by the set of linear equations (12) and (13) is referred to as the max-plus linear system.

As the reader may have noticed, the equations are similar to the state-space representation in modern control theory in conventional algebra.

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C} \cdot \mathbf{x}(t) \end{aligned}$$

With this similarity, several research developments in modern control theory have been applied to max-plus algebraic systems, the details of which will be explained in the following section.

### 3. Literature Review

We introduce several typical research accomplishments with respect to the state-space representation approach in max-plus and Dioid algebras. Roughly speaking, the relevant research may be classified into two types: methodologies and applications. After briefly outlining several research areas, we explain our own research motivation and objectives.

#### 3.1 Methodologies

As mentioned above, the state-space representation in Dioid algebra is similar to the representation in modern control theory in conventional algebra. Thus, several research developments in modern control theory have been applied to Dioid or max-plus algebraic systems, and they now provide several useful and powerful tools for a class of discrete event systems. Typical examples include supervisory control, IMC (Internal Model Control), MPC (Model Predictive Control), and adaptive control and fully described in the current literature.

For instance, the concept of supervisory control is applied in Ramadge & Wonham (1987) and Cofer & Garg (1996). In particular, the latter takes the framework of supervisory control for a timed event graph into account in max-plus algebra. If the specification for a system is given by a set of firing times for transitions, the control specification can be accomplished by delaying the firing times of controllable transitions. This is caused by control signals from the supervisor.

In Boimond & Ferrier (1996), the concept of IMC, often utilised in controller designs for chemical plants, is applied. With these developments, a controller installed in a target

system adjusts completion times to desired times. A general result of this study is that the control inputs for perturbed systems can be made robust.

In Schutter & Boom (2001) and Boom et al. (2007), the concept of MPC has been utilised. MPC determines the control inputs by solving an optimisation problem in which the performance of the system for a finite step is formulated. In addition to MPC, a theory of adaptive control is applied in Schullerus et al. (2006) and Boom et al. (2003). In particular, Boom et al. (2003) realises on-line control by combining a method for system identification and MPC, which they call adaptive MPC. This controller can adjust the states on-line even when the properties of a system are changed unexpectedly.

We can also find other research studies on controller designs for hybrid systems (Heemels et al., 2001) and parameter estimation problems of state-space representations (Schutter et al., 2002).

### 3.2 Applications

Several application fields for practical systems are introduced. A typical field of application is manufacturing systems, as illustrated in the previous section. In modelling these types of systems, feeding times of resource materials and completion times of manufactured parts correspond to input and output variables for the system, respectively. Each process's start and processing times are assigned to internal states and system parameters, respectively.

Similar examples include diagnosis and fault detection for batch-processing lines (Sampath et al., 1996; Schullerus & Krebs, 2001). In such systems, the input times correspond to start times for injection of a substance or solvent, and the output times are equal to completion times for the outflow of the resulting substance. The system parameters are equal to the reaction times, which include the injection and the outflow times. The internal states are start times for the injection or completion times for the outflow.

Several problems in transportation planning using max-plus algebra are reported (Heidergott & Veries, 2001; Moh et al., 2005; Goverde, 2007). These problems can be formulated by setting the system variables in the following manner: For instance, in railway networks, the respective inputs and outputs correspond to departure times from stations of origin, and arrival times at terminals. The system parameters are equivalent to travel times between stations, and the internal states correspond to the departure or arrival times at intermediate stations.

In addition to the studies described above, we can also find developments in TCP flow-control problems arising in the field of communication networks (Baccelli & Hong, 2000).

### 3.3 Problems to be resolved

As introduced above, much attention has been paid to modelling and analysis methods based on Dioid and max-plus algebras. However, there are currently obstacles in this approach to their practical use. In actual systems, there are usually constraints regarding the maximum in-process jobs that can exist within single and between facilities. These are interpreted as capacity constraints. Moreover, occupation times in facilities, processing times for instance, differ for each job in several systems. This situation requires considering additional constraints to disallow overtaking of a previous job or jobs. In Krivulin (1996), a queuing model which can consider the capacity constraints in single facilities was proposed. However, the paper assumes that occupation times are fixed and independent of job



numbers, and no capacity constraint between facilities can be taken into account. Furthermore, based on current methods, deriving the state-space representation is performed manually and ad-hoc, as no systematic and unified method is available. In the light of these difficulties, the following sections propose a systematic framework for deriving the state-space representation. The target systems are allowed to have capacity constraints within single and between facilities, and moreover have varying processing times for each job.

## 4. Considering the Capacity Constraints

We extend the conventional state-space representation in Dioid algebra, and derive a systematic framework for modelling a class of repetitive systems with capacity constraints. Prior to the extension of the state-space representation, we first introduce several operators.

### 4.1 Additional operators

For use in later discussions, we define additional operators and elements. First, we denote the field  $\mathbf{R} \cup \{-\infty\} \cup \{+\infty\}$  by  $\overline{\mathbf{R}}_{\max}$ . For scalar variables  $x, y \in \overline{\mathbf{R}}_{\max}$ , we define the following operators.

$$x \wedge y = \min(x, y), \quad x \setminus y = -x + y$$

The first definition satisfies the commutative law:  $x \wedge y = y \wedge x$ ; in contrast, the second is non-commutative. For the zero element of  $\wedge$ , we define  $\mathbf{T} (= +\infty)$ . This yields  $x \wedge \mathbf{T} = \mathbf{T} \wedge x = x$  and  $x \setminus \mathbf{T} = \mathbf{T}$ . In addition, we enforce the following properties for operator  $\otimes$ :

$$\varepsilon \otimes \mathbf{T} = \mathbf{T} \otimes \varepsilon = \varepsilon \quad (14)$$

based on the axiomatic rules in (7). For operator  $\setminus$ , we define the following operation rules for mathematical convenience:

$$\varepsilon \setminus \varepsilon = \mathbf{T} \setminus \mathbf{T} = \mathbf{T} \quad (15)$$

In conventional algebra, (14) is tantamount to defining the rule:  $(-\infty) + (+\infty) = (+\infty) + (-\infty) = -\infty$ . In contradistinction, (15) corresponds to the rule  $-(-\infty) + (-\infty) = -(+\infty) + (+\infty) = +\infty$ . Both seem to be contradictory in terms of conventional algebraic systems. However, we should note here that these rules are defined exclusively for operators  $\otimes$  and  $\setminus$ , not for  $+$  and  $-$ .

For multiple numbers, if  $x_i \in \overline{\mathbf{R}}_{\max}$ , we simply denote:

$$\bigwedge_{k=1}^l x_k \equiv x_1 \wedge x_2 \wedge \cdots \wedge x_l$$

For matrices  $X, Y \in \overline{\mathbf{R}}_{\max}^{m \times n}$  and  $Z \in \overline{\mathbf{R}}_{\max}^{n \times l}$ , we define the following two operations in analogy to the  $\oplus$  and  $\otimes$  operations.

$$[X \wedge Y]_{ij} = \min([X]_{ij}, [Y]_{ij}), [X \odot Z]_{ij} = \bigwedge_{k=1}^l ([X]_{ik} \setminus [Z]_{kj}) = \min_{k=1, \dots, l} (-[X]_{ik} + [Z]_{kj})$$

For simplicity, several references adopt a different definition for operator  $\odot$ , where  $X \odot Z$  gives the same result as  $X^T \odot Z$  based on the above definition. Nevertheless, we have defined the above rule in an analogous manner to operator  $\otimes$ . In referencing the relevant papers, we recommend verifying its definition first.

For  $X, Y \in \overline{R}_{\max}^{m \times n}$ ,  $Z \in \overline{R}_{\max}^{n \times m}$ ,  $v, w \in \overline{R}_{\max}^n$ , the following properties hold:

$$(X \oplus Y) \odot v = (X \odot v) \wedge (Y \odot v), X \odot (v \wedge w) = (X \odot v) \wedge (X \odot w), \\ Y^T \odot (Z^T \odot v) = (Z \otimes Y)^T \odot v \quad (16)$$

The operators  $\wedge$  and  $\setminus$  also have other interesting and attractive properties that are not used in this chapter. The interested reader is referred to Heidergott et al. (2006) or Baccelli et al. (1992) for details.

## 4.2 Assumptions and notations

Assumptions and notations for the target systems are clarified here. Although we use terms adapted from manufacturing systems, the same concepts can also be applied to other classes of discrete event systems such as transportation systems.

Assume the system has a fork-join structure with  $n$  facilities,  $m$  external inputs, and  $p$  external outputs. Transit times between facilities are initially ignored although they are considered in a later subsection. With respect to order constraints, assume the following are imposed:

- Each job uses all facilities and each is used only once. Thus, the system has an acyclic structure.
- Facilities with predecessors cannot start processing until the process in the preceding facility is finished.
- Facilities that have external inputs cannot start processing until all required resource materials are supplied.
- Facilities that have capacity constraints cannot start processing until the number of in-process jobs in the corresponding region is equal to, or less than, the predetermined value.
- Process start and completion occur sequentially according to job number order in all facilities. In other words, the jobs are processed based on a FIFO (First-In, First-Out) policy.

For the  $k$ -th job in facility  $i$  ( $1 \leq i \leq n$ ), denote the processing time, process start and completion times by  $d_i(k) (\geq 0)$ ,  $[x^s(k)]_i$  and  $[x^c(k)]_i$ , respectively. For external input  $i$  ( $1 \leq i \leq m$ ),  $[u(k)]_i$  represents the material feeding time. For external output  $i$  ( $1 \leq i \leq p$ ),  $[y(k)]_i$  denotes the output time for the product. Subscript suffixes  $E$  and  $L$  are used to express the earliest and latest times.

### 4.3 Forward type representation

We extend the state-space representation (12) and (13). We refer to this type of representation as forward type, by which the earliest start and completion times of the various processes are calculated. The essence of the extension is to take into account constraints with respect to buffer capacities in single and between facilities.

To represent several parameters and constraints such as processing times, precedence relationships, and locations of external input and outputs, we introduce the following matrix parameters  $\mathbf{P}_k, \mathbf{F}, \mathbf{H}^{(h)} \in \mathbf{R}_{\max}^{n \times n}, \mathbf{B} \in \mathbf{R}_{\max}^{n \times m}$  and  $\mathbf{C} \in \mathbf{R}_{\max}^{p \times n}$ :

$$[\mathbf{P}_k]_{ij} = \begin{cases} d_i(k) & \text{if } i = j \\ \varepsilon & \text{if } i \neq j \end{cases}$$

$$[\mathbf{F}]_{ij} = \begin{cases} e & \text{Facility } i \text{ has a preceding facility } j \\ \varepsilon & \text{Facility } i \text{ does not have a preceding facility } j \end{cases}$$

$$[\mathbf{H}^{(h)}]_{ij} = \begin{cases} e & \text{The maximum number of jobs that can exist between facility } i \text{ and its downstream facility } j \text{ is } h \\ \varepsilon & \text{The number of jobs between facilities } i \text{ and } j \text{ is not constrained} \end{cases}$$

$$[\mathbf{B}]_{ij} = \begin{cases} e & \text{Facility } i \text{ has an external input } j \\ \varepsilon & \text{Facility } i \text{ does not have an external input } j \end{cases}$$

$$[\mathbf{C}]_{ij} = \begin{cases} e & \text{External output } i \text{ has a preceding facility } j \\ \varepsilon & \text{External output } i \text{ does not have a preceding facility } j \end{cases}$$

We refer to these matrices as the weight, adjacency, capacity, input and output matrices, respectively. Moreover, for facility  $i$ , denote the list of preceding facilities, external inputs, and downstream facilities with maximum capacity  $h (\geq 1)$  by  $\mathbf{R}_i, \mathbf{Q}$  and  $\mathbf{M}_{ih}$ , respectively. Fig. 2 depicts an image of these symbols.

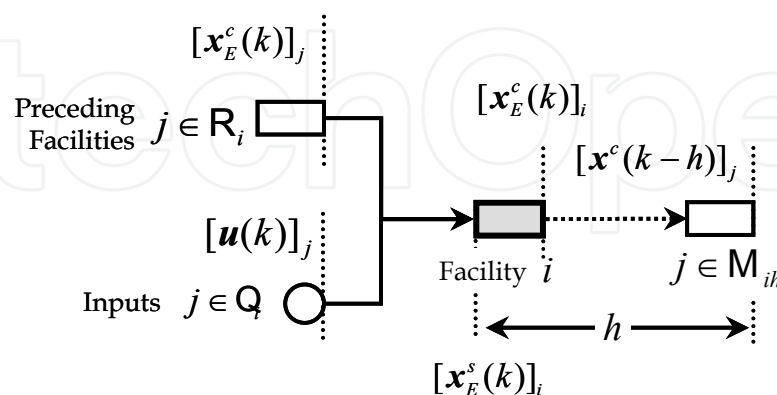


Fig. 2. External inputs and facilities following the  $i$ -th facility

Let us now obtain the earliest process start and completion times in facility  $i$  ( $1 \leq i \leq n$ ). With regard to process completion times, we stipulate that each must be equal or greater than the following two time instants:

- The time at which the processing time  $d_i(k)$  has elapsed from the earliest process start time  $[\mathbf{x}_E^s(k)]_i$ .
- The process completion time of the previous job  $[\mathbf{x}^c(k-1)]_i$ .

The second condition is established by the FIFO policy. Thus, the earliest process completion time, denoted by  $[\mathbf{x}_E^c(k)]_i$ , is formulated using the weight matrix  $\mathbf{P}_k$  as follows:

$$\begin{aligned} [\mathbf{x}_E^c(k)]_i &= ([\mathbf{x}_E^s(k)]_i + d_i(k)) \oplus [\mathbf{x}^c(k-1)]_i = \bigoplus_{j=1}^n ([\mathbf{P}_k]_{ij} \otimes [\mathbf{x}_E^s(k)]_j) \oplus [\mathbf{x}^c(k-1)]_i \\ &= [\mathbf{P}_k \mathbf{x}_E^s(k)]_i \oplus [\mathbf{x}^c(k-1)]_i \end{aligned} \quad (17)$$

Next, we consider the earliest process start time. To begin the process in facility  $i$ , all conditions below must be satisfied.

- All processes in the preceding facilities  $\mathbf{R}_i$  are completed.
- All required materials from the external inputs  $\mathbf{Q}$  are supplied.
- The number of on-going jobs between facilities  $i$  and  $j \in \mathbf{M}_{ih}$  is equal or smaller than  $h$ .
- Processing of the previous job  $k-1$  has begun.

The third condition corresponds to capacity constraints, and the last invokes the FIFO policy. Accordingly, the earliest process start time, denoted by  $[\mathbf{x}_E^s(k)]_i$ , is formulated in the following manner.

$$\begin{aligned} [\mathbf{x}_E^s(k)]_i &= \left( \bigoplus_{j \in \mathbf{R}_i} [\mathbf{x}_E^c(k)]_j \right) \oplus \left( \bigoplus_{j \in \mathbf{Q}} [\mathbf{u}(k)]_j \right) \oplus \left( \bigoplus_{h=1}^H \bigoplus_{j \in \mathbf{M}_{ih}} [\mathbf{x}^c(k-h)]_j \right) \oplus [\mathbf{x}^s(k-1)]_i \\ &= \left( \bigoplus_{j=1}^n [\mathbf{F}]_{ij} \otimes [\mathbf{x}_E^c(k)]_j \right) \oplus \left( \bigoplus_{j=1}^m [\mathbf{B}]_{ij} \otimes [\mathbf{u}(k)]_j \right) \\ &\quad \oplus \left( \bigoplus_{h=1}^H \bigoplus_{j=1}^n [\mathbf{H}^{(h)}]_{ij} \otimes [\mathbf{x}^c(k-h)]_j \right) \oplus [\mathbf{x}^s(k-1)]_i \\ &= [\mathbf{F} \mathbf{x}_E^c(k)]_i \oplus [\mathbf{B} \mathbf{u}(k)]_i \oplus \bigoplus_{h=1}^H [\mathbf{H}^{(h)} \mathbf{x}^c(k-h)]_i \oplus [\mathbf{x}^s(k-1)]_i \end{aligned} \quad (18)$$

$H$  is the maximum buffer size imposed on the system. Noting that (17) and (18) hold true for all  $i$  ( $1 \leq i \leq n$ ), they can be summarised in matrix form as follows:

$$\begin{bmatrix} \mathbf{x}_E^s(k) \\ \mathbf{x}_E^c(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon} & \mathbf{F} \\ \mathbf{P}_k & \boldsymbol{\varepsilon} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{x}_E^s(k) \\ \mathbf{x}_E^c(k) \end{bmatrix} \oplus \begin{bmatrix} \mathbf{B} \\ \boldsymbol{\varepsilon} \end{bmatrix} \otimes \mathbf{u}(k) \\ \oplus \bigoplus_{h=1}^H \begin{bmatrix} \boldsymbol{\varepsilon} & \mathbf{H}^{(h)} \\ \boldsymbol{\varepsilon} & \boldsymbol{\varepsilon} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{x}_E^s(k-h) \\ \mathbf{x}_E^c(k-h) \end{bmatrix} \oplus \begin{bmatrix} \mathbf{e} & \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} & \mathbf{e} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{x}^s(k-1) \\ \mathbf{x}^c(k-1) \end{bmatrix}$$

This can be simply represented as follows:

$$\mathbf{x}_E(k) = \bar{\mathbf{F}}_k \mathbf{x}_E(k) \oplus \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k) \quad (19)$$

where:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}^s(k) \\ \mathbf{x}^c(k) \end{bmatrix}, \quad \bar{\mathbf{F}}_k = \begin{bmatrix} \boldsymbol{\varepsilon} & \mathbf{F} \\ \mathbf{P}_k & \boldsymbol{\varepsilon} \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \boldsymbol{\varepsilon} \end{bmatrix}, \quad \bar{\mathbf{H}}^{(1)} = \begin{bmatrix} \mathbf{e} & \mathbf{H}^{(1)} \\ \boldsymbol{\varepsilon} & \mathbf{e} \end{bmatrix}, \quad \bar{\mathbf{H}}^{(h)} = \begin{bmatrix} \boldsymbol{\varepsilon} & \mathbf{H}^{(h)} \\ \boldsymbol{\varepsilon} & \boldsymbol{\varepsilon} \end{bmatrix} \quad h \geq 2 \quad (20)$$

We note here that (19) is an implicit expression for  $\mathbf{x}_E(k)$ . Thus, by substituting the entire right-hand-side of (19) with  $\mathbf{x}_E(k)$  in the first term of the right-hand-side, we obtain the following relationship:

$$\begin{aligned} \mathbf{x}_E(k) &= \bar{\mathbf{F}}_k \left( \bar{\mathbf{F}}_k \mathbf{x}_E(k) \oplus \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k) \right) \oplus \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k) \\ &= \bar{\mathbf{F}}_k^2 \mathbf{x}_E(k) \oplus (\mathbf{e} \oplus \bar{\mathbf{F}}_k) \left[ \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k) \right] \end{aligned}$$

Furthermore, by repeating this transformation, we obtain:

$$\begin{aligned} \mathbf{x}_E(k) &= \bar{\mathbf{F}}_k^3 \mathbf{x}_E(k) \oplus (\mathbf{e} \oplus \bar{\mathbf{F}}_k \oplus \bar{\mathbf{F}}_k^2) \left[ \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k) \right] \\ &\dots = \bar{\mathbf{F}}_k^l \mathbf{x}_E(k) \oplus (\mathbf{e} \oplus \bar{\mathbf{F}}_k \oplus \bar{\mathbf{F}}_k^2 \oplus \dots \oplus \bar{\mathbf{F}}_k^{l-1}) \left[ \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k) \right] \end{aligned} \quad (21)$$

With regard to  $\bar{\mathbf{F}}_k$ , the following relationship holds:

$$\begin{aligned} \bar{\mathbf{F}}_k^2 &= \begin{bmatrix} \mathbf{F}\mathbf{P}_k & \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} & \mathbf{P}_k\mathbf{F} \end{bmatrix}, \quad \bar{\mathbf{F}}_k^3 = \begin{bmatrix} \boldsymbol{\varepsilon} & (\mathbf{F}\mathbf{P}_k)\mathbf{F} \\ (\mathbf{P}_k\mathbf{F})\mathbf{P}_k & \boldsymbol{\varepsilon} \end{bmatrix}, \quad \bar{\mathbf{F}}_k^4 = \begin{bmatrix} (\mathbf{F}\mathbf{P}_k)^2 & \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} & (\mathbf{P}_k\mathbf{F})^2 \end{bmatrix}, \quad \dots, \\ \bar{\mathbf{F}}_k^{2l-1} &= \begin{bmatrix} \boldsymbol{\varepsilon} & (\mathbf{F}\mathbf{P}_k)^l \mathbf{F} \\ (\mathbf{P}_k\mathbf{F})^l \mathbf{P}_k & \boldsymbol{\varepsilon} \end{bmatrix}, \quad \bar{\mathbf{F}}_k^{2l} = \begin{bmatrix} (\mathbf{F}\mathbf{P}_k)^l & \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} & (\mathbf{P}_k\mathbf{F})^l \end{bmatrix} \end{aligned}$$

In addition, there is an instance  $s$  ( $2 \leq s \leq n$ ) that satisfies:

$$(\mathbf{FP}_k)^{s-1} \neq \varepsilon, (\mathbf{P}_k \mathbf{F})^{s-1} \neq \varepsilon, (\mathbf{FP}_k)^s = (\mathbf{P}_k \mathbf{F})^s = \varepsilon$$

which is dependent on the precedence relation of the system. With the help of this property, (21) is finally transformed into:

$$\mathbf{x}_E(k) = \bar{\mathbf{F}}_k^* \left[ \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k) \right] \quad (22)$$

where:

$$\begin{aligned} \bar{\mathbf{F}}_k^* &= \mathbf{e} \oplus \bar{\mathbf{F}} \oplus \dots \oplus \bar{\mathbf{F}}^{2s-1} = \begin{bmatrix} (\mathbf{FP}_k)^* & (\mathbf{FP}_k)^* \mathbf{F} \\ (\mathbf{P}_k \mathbf{F})^* \mathbf{P}_k & (\mathbf{P}_k \mathbf{F})^* \end{bmatrix}, \\ (\mathbf{FP}_k)^* &= \mathbf{e} \oplus (\mathbf{FP}_k) \oplus \dots \oplus (\mathbf{FP}_k)^{s-1}, (\mathbf{P}_k \mathbf{F})^* = \mathbf{e} \oplus (\mathbf{P}_k \mathbf{F}) \oplus \dots \oplus (\mathbf{P}_k \mathbf{F})^{s-1} \end{aligned}$$

Superscript  $*$  refers to the Kleene star (Heidergott et al., 2006), a well-known concept in the field of information theory. The original definition assumes the infinite summation over sequential powers of a given matrix:

$$\mathbf{X}^* = \bigoplus_{l=0}^{\infty} \mathbf{X}^l = \mathbf{e} \oplus \mathbf{X} \oplus \mathbf{X}^2 \oplus \dots$$

If there is an instance  $s$  such that  $\mathbf{X}^{s-1} \neq \varepsilon$  and  $\mathbf{X}^s = \varepsilon$ ,  $\mathbf{X}$  is said to be nilpotent and the above operation reduces to a finite sum of powers of  $\mathbf{X}$ . The adjacency matrix of target systems is a case in point. Several efficient computation methods for the Kleene star have been proposed. See Goto & Takahashi (2009) for details.

Moreover, we note here that  $\mathbf{P}_k (\mathbf{FP}_k)^* = (\mathbf{P}_k \mathbf{F})^* \mathbf{P}_k$  holds. This means that  $(\mathbf{FP}_k)^*$  and  $(\mathbf{P}_k \mathbf{F})^*$  are related as follows:

$$[(\mathbf{FP}_k)^*]_{ij} + d_i(k) = [(\mathbf{P}_k \mathbf{F})^*]_{ij} + d_j(k) \quad (23)$$

Thus, once either  $(\mathbf{FP}_k)^*$  or  $(\mathbf{P}_k \mathbf{F})^*$  has been calculated, the other can be calculated with low computation load.

Next, we consider the earliest output time. For external output  $i$ , let us denote the list of preceding facilities by  $\mathbf{T}_i$ . Then, the output time must be equal or greater than the maximisation of the process completion times in these facilities. Thus, the earliest output time in external output  $i$  ( $1 \leq i \leq p$ ) can be expressed as:

$$[\mathbf{y}_E(k)]_i = \left( \bigoplus_{j \in \mathbf{T}_i} [\mathbf{x}^c(k)]_j \right) = \left( \bigoplus_{j=1}^p [\mathbf{C}]_{ij} \otimes [\mathbf{x}^c(k)]_j \right) = [\mathbf{C} \mathbf{x}^c(k)]_i$$



Since this holds true for all  $i$  ( $1 \leq i \leq p$ ),  $y_E(k) = Cx^c(k)$  is obtained. Moreover, this can also be represented as the following using the same state variable as appears in (22):

$$y_E(k) = \bar{C}x(k) \quad (24)$$

where:

$$\bar{C} = [\varepsilon \ C] \quad (25)$$

Equations (22) and (24) are extended versions of the state and output equations, respectively.

#### 4.4 Backward type State-space representation

We derive a backward state-space representation taking capacity constraints into account. The same matrix parameters,  $P_k$ ,  $F$ ,  $H^{(h)} \in R_{\max}^{n \times n}$ ,  $B \in R_{\max}^{n \times m}$  and  $C \in R_{\max}^{p \times n}$ , are used as in the previous subsection. Fig. 3 depicts the relevant constraints regarding facility  $i$  ( $1 \leq i \leq n$ ). With respect to facility  $i$ ,  $S_i$  and  $P_i$  represent the number of succeeding facilities and attached external outputs, respectively. Suppose there is a constraint for the maximum number of jobs between the process completion point in facility  $i$  and the process starting point in upstream facility  $j$ , and denote the collection of facilities  $j$  by  $N_{ih}$  if its corresponding number is  $h$ . For the  $k$ -th job in facility  $i$ , represent the latest process starting and completion times as  $[x_L^s(k)]_i$  and  $[x_L^c(k)]_i$ , respectively.

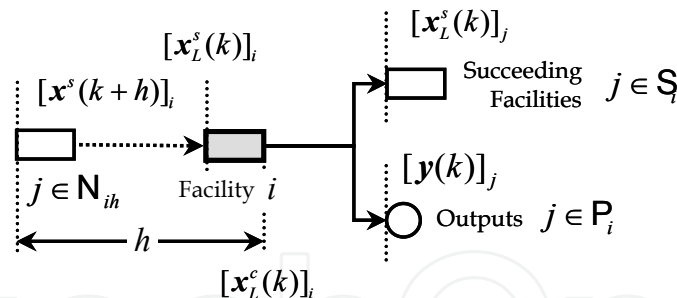


Fig. 3. External output and facilities following the  $i$ -th facility

The completion time of the  $k$ -th job in facility  $i$  is equal to, or earlier than, the following four times:

- The latest start time  $[x_L^s(k)]_j$  in succeeding facilities  $j \in S_i$
- Output time  $[y(k)]_j$  to external output  $j \in P_i$
- The start time  $[x^s(k+h)]_j$  of the  $(k+h)$ -th job in upstream facilities  $j \in N_{ih}$
- The completion time  $[x^c(k+1)]_i$  of the subsequent job

Accordingly, the latest completion time in facility  $i$  can be formulated as follows:

$$\begin{aligned}
 [\mathbf{x}_L^c(k)]_i &= \left( \bigwedge_{j \in \mathbb{S}} [\mathbf{x}_L^s(k)]_j \right) \wedge \left( \bigwedge_{j \in \mathbb{P}_i} [\mathbf{y}(k)]_j \right) \wedge \left( \bigwedge_{h=1}^H \bigwedge_{j \in \mathbb{N}_{ih}} [\mathbf{x}^s(k+h)]_j \right) \wedge [\mathbf{x}^c(k+1)]_i \\
 &= \left( \bigwedge_{j=1}^n [\mathbf{F}^T]_{ij} \setminus [\mathbf{x}_L^s(k)]_j \right) \wedge \left( \bigwedge_{j=1}^p [\mathbf{C}^T]_{ij} \setminus [\mathbf{y}(k)]_j \right) \\
 &\quad \wedge \left( \bigwedge_{h=1}^H \bigwedge_{j=1}^n [\mathbf{H}^{(h)T}]_{ij} \setminus [\mathbf{x}^s(k+h)]_j \right) \wedge [\mathbf{x}^c(k+1)]_i \\
 &= [\mathbf{F}^T \odot \mathbf{x}_L^s(k)]_i \wedge [\mathbf{C}^T \odot \mathbf{y}(k)]_i \wedge \bigwedge_{h=1}^H [\mathbf{H}^{(h)T} \odot \mathbf{x}^s(k+h)]_i \wedge [\mathbf{x}^c(k+1)]_i
 \end{aligned} \tag{26}$$

Moreover, the process starting time of the  $k$ -th job in facility  $i$  is equal to, or earlier than:

- The time at which  $d_i(k)$  is subtracted from the latest completion time in the corresponding facility.
- The start time of the next job  $[\mathbf{x}^s(k+1)]_i$ .

Thus, the latest start time for processing can be formulated as follows:

$$\begin{aligned}
 [\mathbf{x}_L^s(k)]_i &= ([\mathbf{x}_L^c(k)]_i - d_i(k)) \wedge [\mathbf{x}^s(k+1)]_i \\
 &= \bigwedge_{j=1}^n [\mathbf{P}_k^T]_{ij} \setminus [\mathbf{x}_L^c(k)]_j \wedge [\mathbf{x}^s(k+1)]_i = [\mathbf{P}_k^T \odot \mathbf{x}_L^c(k)]_i \wedge [\mathbf{x}^s(k+1)]_i
 \end{aligned} \tag{27}$$

Equations (26) and (27) hold true for all  $i$  ( $1 \leq i \leq n$ ), and can be summarised in matrix form as follows:

$$\begin{aligned}
 \begin{bmatrix} \mathbf{x}_L^s(k) \\ \mathbf{x}_L^c(k) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\varepsilon} & \mathbf{P}_k^T \\ \mathbf{F}^T & \boldsymbol{\varepsilon} \end{bmatrix} \odot \begin{bmatrix} \mathbf{x}_L^s(k) \\ \mathbf{x}_L^c(k) \end{bmatrix} \wedge \begin{bmatrix} \boldsymbol{\varepsilon} \\ \mathbf{C}_0^T \end{bmatrix} \odot \mathbf{y}(k) \\
 &\quad \wedge \bigwedge_{h=1}^Q \begin{bmatrix} \boldsymbol{\varepsilon} & \boldsymbol{\varepsilon} \\ \mathbf{H}^{(h)T} & \boldsymbol{\varepsilon} \end{bmatrix} \odot \begin{bmatrix} \mathbf{x}_L^s(k+h) \\ \mathbf{x}_L^c(k+h) \end{bmatrix} \wedge \begin{bmatrix} \mathbf{e} & \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} & \mathbf{e} \end{bmatrix} \odot \begin{bmatrix} \mathbf{x}^s(k+1) \\ \mathbf{x}^c(k+1) \end{bmatrix}
 \end{aligned}$$

Moreover, using the augmented matrices in (20) and (25), the following simplified expression is obtained:

$$\mathbf{x}_L(k) = \overline{\mathbf{F}}_k^T \odot \mathbf{x}_L(k) \wedge \bigwedge_{h=1}^H \overline{\mathbf{H}}_0^{(h)T} \odot \mathbf{x}(k+h) \wedge \overline{\mathbf{C}}_0^T \odot \mathbf{y}(k) \tag{28}$$

Equation (28) is an implicit form of  $\mathbf{x}_L(k)$ . Iteratively substituting the entire right side of (28) with the first term and using the relational expressions in (16), equation (28) is transformed into the following explicit form:

$$\begin{aligned} \mathbf{x}_L(k) &= \overline{\mathbf{F}}_k^{2^T} \odot \mathbf{x}_L(k) \wedge (\mathbf{e} \oplus \overline{\mathbf{F}}_k)^T \odot \left[ \bigwedge_{h=1}^H \overline{\mathbf{H}}^{(h)T} \odot \mathbf{x}(k+h) \wedge \overline{\mathbf{C}}^T \odot \mathbf{y}(k) \right] \\ &= \dots = \overline{\mathbf{F}}_k^{*T} \odot \left[ \bigwedge_{h=1}^H \overline{\mathbf{H}}^{(h)T} \odot \mathbf{x}(k+h) \wedge \overline{\mathbf{C}}^T \odot \mathbf{y}(k) \right] \end{aligned} \quad (29)$$

Fig. 4 depicts the relationships regarding external input  $i$  relevant to calculating the latest input time.  $\mathbf{W}_i$  is a collection of succeeding facilities attached to external input  $i$ . Since the start time for the  $k$ -th job in succeeding facility  $j \in \mathbf{W}_i$  is  $[\mathbf{x}^s(k)]_j$ , the latest feed time for the corresponding job  $[\mathbf{u}_L(k)]_i$  can be determined as follows:

$$[\mathbf{u}_L(k)]_i = \bigwedge_{j \in \mathbf{W}_i} ([\mathbf{x}^s(k)]_j) = \bigwedge_{j=1}^n ([\mathbf{B}^T]_{ij} \setminus [\mathbf{x}^s(k)]_j) = [\mathbf{B}^T \odot \mathbf{x}^s(k)]_i$$

This holds true for all  $i$  ( $1 \leq i \leq m$ ), and can also be expressed using the same state vector as (22), thus:

$$\mathbf{u}_L(k) = \overline{\mathbf{B}}^T \odot \mathbf{x}(k) \quad (30)$$

From the above we obtain the state equation (29) and output equation (30) which represents the latest possible times for the  $k$ -th job.

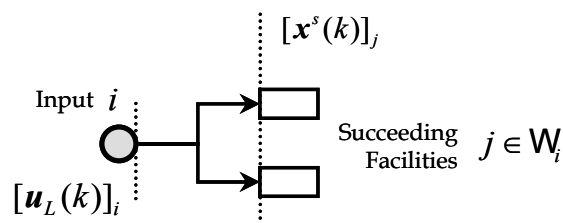


Fig. 4. Facilities following the  $i$ -th external input

#### 4.5 The parameter matrix of the capacity constraint

This subsection concentrates on a method for generating matrices  $\mathbf{H}^{(h)}$  that specify buffer capacities between facilities. Such a method is required to provide  $H$  matrices for deriving state equations, that may be complicated if they are specified individually. Hence, we provide a single matrix  $\mathbf{G}$  to represent all capacity constraints.

$$[\mathbf{G}]_{ij} = \begin{cases} e: \text{The maximum number of jobs that can exist between facility } i \text{ and its downstream facility } j \text{ is } h \\ \varepsilon: \text{There is no constraint on the number of jobs from } i \text{ to } j \end{cases}$$

The downstream facility  $j$  may include facility  $i$  itself, namely  $i = j$ . For this definition, the following relation holds true:

$$\bigoplus_{j=1}^n \bigoplus_{i=1}^n [\mathbf{G}]_{ij} = H$$

Matrices  $\mathbf{H}^{(h)}$  can be generated by applying the following rule for all  $h$  ( $1 \leq h \leq Q$ ).

$$[\mathbf{H}^{(h)}]_{ij} = \begin{cases} e & \text{if } [\mathbf{G}]_{ij} = h \\ \varepsilon & \text{otherwise} \end{cases}$$

For systems in which the maximum buffer is one for a single facility and infinite between adjacent facilities, the parameter matrix is  $\mathbf{G} = \mathbf{e}$ . Moreover, the definition of  $\mathbf{G}$  yields:

$$\text{If } [\mathbf{G}]_{ij} \neq \varepsilon, [\mathbf{G}]_{ji} = \varepsilon$$

for all  $i$  and  $j$  ( $i \neq j$ ).

4.6 Consideration of transit times

From here on the transit times between adjacent facilities that were ignored to this point are taken into account. First, let us consider a case where transit time is constant and does not depend on the job number  $k$ . Since jobs do not overtake each other during transits here, no additional order constraints need be considered. To take transit times into account, we need only set the  $(i, j)$ -th element of the adjacent matrix that holds  $[\mathbf{F}]_{ij} = e$  for the corresponding transit time. Alternatively, if the transit times between facilities depend on the job number,  $k$ , additional order constraints should be considered. In this case, we can install an imaginary facility between adjacent facilities. Consider the case presented in Fig. 5 as an example. Assume the transit time from facility  $b$  to  $a$  is dependent on the job number,  $k$ , and let this time be represented by  $\tau_{ab}(k)$ . Here the order constraint is forced to disallow overtaking between successive jobs. Thus, we can install an imaginary facility  $s$  between facilities  $b$  and  $a$ , whose occupation time for the  $k$ -th job is  $\tau_{ab}(k)(\equiv d_s(k))$ . In addition to the installation of the imaginary facility, we can update the adjacency matrix  $\mathbf{F}$ . The original matrix follows the next relationship:

$$[\mathbf{F}]_{ab} = e (\equiv [\mathbf{F}^{(0)}]_{ab} \in \mathbf{R}_{\max}^{n \times n})$$

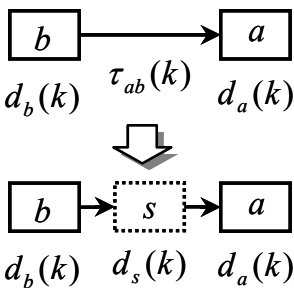


Fig. 5. Installation of an imaginary facility

Through the installation of facility  $s$ , the modified adjacency matrix  $\hat{\mathbf{F}}^{(1)} \in \mathbf{R}_{\max}^{(n+1) \times (n+1)}$  satisfies the following properties:

$$\hat{\mathbf{F}}^{(1)} = \begin{bmatrix} \mathbf{F}^{(1)} & \mathbf{v}_a^{(1)} \\ \mathbf{v}_b^{(1)T} & \varepsilon \end{bmatrix}, [\mathbf{F}^{(1)}]_{ij} = \begin{cases} \varepsilon & \text{if } i = a \text{ and } j = b \\ [\mathbf{F}^{(0)}]_{ij} & \text{otherwise} \end{cases}$$

where

$$[\mathbf{v}_a^{(1)}]_b = \begin{cases} e & \text{if } a = b \\ \varepsilon & \text{if } a \neq b \end{cases}, \mathbf{v}_a^{(1)} \in \mathbf{R}_{\max}^{n \times n}$$

Moreover, the matrix parameter  $\mathbf{P}_k$  is modified in the following manner:

$$\hat{\mathbf{P}}_k^{(1)} = \begin{bmatrix} \mathbf{P}_k^{(1)} & \varepsilon \\ \varepsilon & d_s(k) \end{bmatrix}$$

A new adjacency matrix can be generated using this procedure for all paths on which the transit time is dependent on the job number  $k$ . Let the number of installed imaginary facilities be  $g$ , and the modified adjacency matrix be denoted by  $\hat{\mathbf{F}}$ . Then, the remaining representation matrices are modified as follows:

$$\hat{\mathbf{H}}^{(1)} = \begin{bmatrix} \mathbf{H}^{(h)} & \varepsilon \\ \varepsilon & e \end{bmatrix}, \hat{\mathbf{H}}^{(h)} = \begin{bmatrix} \mathbf{H}^{(h)} & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix} (h \geq 2), \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \varepsilon \end{bmatrix}, \hat{\mathbf{C}} = [\mathbf{C} \quad \varepsilon]$$

#### 4.7 Duality of state-space representation

This subsection examines the duality of the derived state-space representations. This duality is understood in a stricter manner than simply similarity of representation, as has been discussed in the previous research.

Goto *et al.* (2007) has focused on systems in which buffer capacities are one in a single facility and infinite between adjacent facilities, which is a narrower class than this chapter handles. This class does not require consideration of order constraints since jobs cannot overtake each other even for event-varying systems. This means either  $[\mathbf{x}^{\{s,c\}}(k-1)]_i$  does not appear in (17) and (18) and  $[\mathbf{x}^{\{s,c\}}(k+1)]_i$  does not appear in (26) and (27), and they form sets of closed equations regarding  $\mathbf{x}_E^c(k)$  and  $\mathbf{x}_L^s(k)$ , respectively. These equations can be represented in a form whose relationship is similar to dual systems in modern control theory. A primary advantage of this duality is that the same system matrices can be used for both forward and backward types, and the calculation time can be reduced accordingly. This reduction is effective for on-line operation especially for large-scale systems.

It is important, at this point, to remember the main concern of this chapter. Since buffer sizes must be considered flexibly, order constraints should be taken into account to disallow overtaking between jobs. This yields a closed equation for  $\mathbf{x}_E^c(k)$  or  $\mathbf{x}_L^s(k)$  that cannot be

formulated as a forward type. The same situation holds true for the representation of backward type state equation. However, if we compose the augmented state-vector  $\mathbf{x}_E(k)$ , equations (17), (18), (29) and (30) can be represented as dual form. This means that all required schedules can be calculated using only four representation matrices,  $\bar{\mathbf{F}}_k^*$ ,  $\bar{\mathbf{H}}^{(h)}$ ,  $\bar{\mathbf{B}}$  and  $\bar{\mathbf{C}}$ . Among these matrices, only  $\bar{\mathbf{F}}_k^*$  depends on the job number. Thus, the question of how to calculate this matrix efficiently especially in event-varying systems becomes central. Due to the composition of the augmented state-space representations, the number of elements in the system matrix is quadrupled. However, as mentioned in (23), once  $(\mathbf{P}_k \mathbf{F}_0)^*$  is calculated, the remaining three blocks of  $\bar{\mathbf{F}}_k^*$  can be calculated by simple algebraic operations. Accordingly, we know that the derived augmented state-space representations are effective, especially in on-line scheduling problems that require calculation of both earliest and latest times.

## 5. Rescheduling

When job parameters are changed after a job has commenced it becomes very important to be able to predict scheduling for remaining jobs in an on-line scheduling system. Typical examples of such changes in parameters are a delay in processing or tardiness in material feeding. Thus, this section considers a rescheduling method for the extended state-space representation.

### 5.1 Forward type

Assume that the system parameters or state variables in previous jobs changed after job commencement, and let the updated values be denoted by appending a tilde symbol  $[\sim]$  in the following manner:

$$\tilde{\mathbf{P}}_k, \tilde{\mathbf{u}}(k), \tilde{\mathbf{x}}(k-h) \quad (1 \leq h \leq Q)$$

Moreover, suppose the  $i$ -th element of the state vector for job number  $k$  has changed as follows:

$$[\tilde{\mathbf{x}}^{(+0)}(k)]_i$$

It is possible that the number of changed elements  $i$  is greater than one. Set  $\varepsilon$  for elements whose corresponding times are to be recalculated. The superscript  $(+0)$  stands for the initial value for the iterative calculation. Equations (17) and (18) are formulated to model the propagation of the earliest starting time downstream. By tracking (22), the earliest start time one-step downstream can be determined as follows:

$$\tilde{\mathbf{x}}_E^{(1)}(k) = \tilde{\mathbf{F}}_k \tilde{\mathbf{x}}^{(+0)}(k) \oplus \bigoplus_{h=1}^H \bar{\mathbf{H}}^{(h)} \tilde{\mathbf{x}}(k-h) \oplus \bar{\mathbf{B}} \tilde{\mathbf{u}}(k) \quad (31)$$



where:

$$\tilde{\mathbf{F}}_k = \begin{bmatrix} \boldsymbol{\varepsilon} & \mathbf{F} \\ \tilde{\mathbf{P}}_k & \boldsymbol{\varepsilon} \end{bmatrix} \quad (32)$$

Note that the element number for (31) is abbreviated for simplicity, it holds true only for elements one-step downstream from the altered facility. Repeating the same procedure downstream, the updated earliest time  $j$ -steps downstream can be obtained thus:

$$\begin{aligned} \tilde{\mathbf{x}}_E^{(j)}(k) &= \tilde{\mathbf{F}}_k \tilde{\mathbf{x}}^{(j-1)}(k) \oplus \bigoplus_{h=1}^H \overline{\mathbf{H}}^{(h)} \tilde{\mathbf{x}}(k-h) \oplus \overline{\mathbf{B}} \tilde{\mathbf{u}}(k) \\ &= \tilde{\mathbf{F}}_k^2 \tilde{\mathbf{x}}^{(j-2)}(k) \oplus (\mathbf{e} \oplus \tilde{\mathbf{F}}_k) \left[ \bigoplus_{h=1}^H \overline{\mathbf{H}}^{(h)} \tilde{\mathbf{x}}(k-h) \oplus \overline{\mathbf{B}} \tilde{\mathbf{u}}(k) \right] \\ \dots &= \tilde{\mathbf{F}}_k^j \tilde{\mathbf{x}}^{(+0)}(k) \oplus (\mathbf{e} \oplus \dots \oplus \tilde{\mathbf{F}}_k^{j-1}) \left[ \bigoplus_{h=1}^H \overline{\mathbf{H}}^{(h)} \tilde{\mathbf{x}}(k-h) \oplus \overline{\mathbf{B}} \tilde{\mathbf{u}}(k) \right] \end{aligned}$$

Calculate this for all  $j$  ( $1 \leq j \leq 2 \cdot s$ ), and use the next trivial relationship:

$$\tilde{\mathbf{x}}_E^{(+0)}(k) = \tilde{\mathbf{x}}^{(+0)}(k)$$

By using the  $\oplus$  operation for these equations, the following expression can be obtained:

$$\tilde{\mathbf{x}}_E(k) \equiv \tilde{\mathbf{x}}_E^{(+0)}(k) \oplus \tilde{\mathbf{x}}_E^{(1)}(k) \oplus \dots \oplus \tilde{\mathbf{x}}_E^{(2 \cdot s)}(k) = \tilde{\mathbf{F}}_k^* \left[ \tilde{\mathbf{x}}^{(+0)}(k) \oplus \bigoplus_{h=1}^H \overline{\mathbf{H}}^{(h)} \tilde{\mathbf{x}}(k-h) \oplus \overline{\mathbf{B}} \tilde{\mathbf{u}}(k) \right] \quad (33)$$

Equation (33) is a general form of forward state equation that is applicable even when the states are changed after the commencement of job  $k$ . If all elements of the state variables require recalculating, the following relation holds:

$$[\tilde{\mathbf{x}}^{(+0)}(k)]_i = \boldsymbol{\varepsilon} \text{ for all } (1 \leq i \leq 2 \cdot n)$$

In this case, equation (33) is equivalent to (22).

Furthermore, consider a particular case that only contains delays in the initial schedule that occur between facilities. For initial values of the state vector, set the latest values for elements in which delays occurred and keep the initial values for other elements. Using these settings, the following relationship holds:

$$\tilde{\mathbf{x}}^{(+0)}(k) \geq \mathbf{x}_E(k)$$

Hence, equation (33) can be simplified as follows:

$$\tilde{\mathbf{x}}_E(k) = \tilde{\mathbf{F}}_k^* \left[ \tilde{\mathbf{x}}^{(+0)}(k) \oplus \bigoplus_{h=1}^H \overline{\mathbf{H}}^{(h)} \tilde{\mathbf{x}}(k-h) \oplus \overline{\mathbf{B}} \tilde{\mathbf{u}}(k) \right] = \tilde{\mathbf{F}}_k^* [\tilde{\mathbf{x}}^{(+0)}(k) \oplus \mathbf{x}_E(k)] = \tilde{\mathbf{F}}_k^* \tilde{\mathbf{x}}^{(+0)}(k) \quad (34)$$

Equation (34) indicates that we can reschedule by performing only the  $\tilde{\mathbf{F}}_k^* \otimes$  operation on the updated state vector  $\tilde{\mathbf{x}}^{(+0)}(k)$ , in cases where only delays from the initial schedule occurred. This expression is much simpler than (33), and provides an easy-to-use method in on-line scheduling, for instance, real-time progress management.

## 5.2 Backward type

Backward states can be handled using a method analogous to that discussed in the previous section. Assume that the system parameters and state variables are changed after the commencement of the  $k$ -th job in the following way:

$$\tilde{\mathbf{P}}_k, \tilde{\mathbf{y}}(k), \tilde{\mathbf{x}}(k+h) \quad (1 \leq h \leq Q)$$

For both the start and completion of the  $k$ -th job, suppose the  $i$ -th element of the state vector is changed as follows:

$$[\tilde{\mathbf{x}}^{(-0)}(k)]_i$$

There may be multiple corresponding elements for  $i$ , and set  $\mathsf{T}$  for elements whose values are to be recalculated. The superscript  $(-0)$  represents the initial value for an iterative calculation. Equations (26) and (27) are formulated to characterise the upstream propagation of the latest times. In a similar way to (28), the latest time one-step upstream can be formulated in the following manner:

$$\tilde{\mathbf{x}}_L^{(1)}(k) = \tilde{\mathbf{F}}_k^T \odot \tilde{\mathbf{x}}^{(-0)}(k) \wedge \bigwedge_{h=1}^H \overline{\mathbf{H}}^{(h)T} \odot \tilde{\mathbf{x}}(k+h) \wedge \overline{\mathbf{C}}^T \odot \tilde{\mathbf{y}}(k) \quad (35)$$

where  $\tilde{\mathbf{F}}_k$  is the same as (32). Equation (35) holds true only for elements one-step upstream from the altered facility. Repeat the same procedure moving upstream, to obtain the latest time for  $J$ -steps upstream. An iterative substitution obtains the following:

$$\begin{aligned} \tilde{\mathbf{x}}_L^{(j)}(k) &= (\tilde{\mathbf{F}}_k^j)^T \odot \tilde{\mathbf{x}}^{(j-1)}(k) \wedge (\mathbf{e} \oplus \tilde{\mathbf{F}}_k)^T \odot \left[ \bigwedge_{h=1}^H \overline{\mathbf{H}}^{(h)T} \odot \tilde{\mathbf{x}}(k+h) \wedge \overline{\mathbf{C}}^T \odot \tilde{\mathbf{y}}(k) \right] \\ &\dots = (\tilde{\mathbf{F}}_k^j)^T \odot \tilde{\mathbf{x}}^{(-0)}(k) \wedge (\mathbf{e} \oplus \dots \oplus \tilde{\mathbf{F}}_k^{j-1})^T \odot \left[ \bigwedge_{h=1}^H \overline{\mathbf{H}}^{(h)T} \odot \tilde{\mathbf{x}}(k+h) \wedge \overline{\mathbf{C}}^T \odot \tilde{\mathbf{y}}(k) \right] \end{aligned}$$

Calculate this for all  $j$  ( $1 \leq j \leq 2 \cdot s$ ), and use the next trivial relationship:

$$\tilde{\mathbf{x}}_L^{(-0)}(k) = \tilde{\mathbf{x}}^{(-0)}(k)$$

The following expression is obtained by performing the  $\wedge$  operation on all the resulting equations:

$$\begin{aligned} \tilde{\mathbf{x}}_L(k) &\equiv \tilde{\mathbf{x}}_L^{(-0)}(k) \wedge \tilde{\mathbf{x}}_L^{(1)}(k) \wedge \cdots \wedge \tilde{\mathbf{x}}_L^{(2 \cdot s)}(k) \\ &= (\tilde{\mathbf{F}}_k^*)^T \odot \left[ \tilde{\mathbf{x}}^{(-0)}(k) \wedge \bigwedge_{h=1}^H \overline{\mathbf{H}}^{(h)T} \odot \tilde{\mathbf{x}}(k+h) \wedge \overline{\mathbf{C}}^T \odot \tilde{\mathbf{y}}(k) \right] \end{aligned} \quad (36)$$

Equation (36) is a general backward type state equation that is applicable even if states are changed after commencement of the  $k$ -th job. If all elements are to be calculated, it follows that:

$$[\tilde{\mathbf{x}}^{(-0)}(k)]_i = \mathbf{T} \text{ for all } (1 \leq i \leq 2 \cdot n)$$

and (36) is equivalent to (29).

Moreover, consider a particular case where the initial schedule is moved forward after job commencement. For the initial values of the state vector, set the updated values for elements whose schedules have been put forward, and keep the original values for the other elements. These settings lead to:

$$\tilde{\mathbf{x}}^{(-0)}(k) \leq \mathbf{x}_L(k)$$

Thus, equation (36) can be simplified to:

$$\begin{aligned} \tilde{\mathbf{x}}_L(k) &= (\tilde{\mathbf{F}}_k^*)^T \odot \left[ \tilde{\mathbf{x}}^{(-0)}(k) \wedge \bigwedge_{h=1}^H \overline{\mathbf{H}}^{(h)T} \odot \tilde{\mathbf{x}}(k+h) \wedge \overline{\mathbf{C}}^T \odot \tilde{\mathbf{y}}(k) \right] \\ &= (\tilde{\mathbf{F}}_k^*)^T \odot [\tilde{\mathbf{x}}^{(-0)}(k) \wedge \mathbf{x}_L(k)] = (\tilde{\mathbf{F}}_k^*)^T \odot \tilde{\mathbf{x}}^{(-0)}(k) \end{aligned} \quad (37)$$

Equation (37) indicates that if the schedule is moved up from the original only the  $(\tilde{\mathbf{F}}_k^*)^T \odot$  operation is required on the updated state vector  $\tilde{\mathbf{x}}^{(-0)}(k)$  for rescheduling. This relationship provides a simpler method than (36).

We now have two state-space representations for event-varying systems with capacity constraints for both forward and backward state spaces.

6. Numerical Experiment

We present an applied example of the proposed method for a simple system. Fig. 6 shows a manufacturing system with two-inputs, one-output and four-facilities. F1-F4 represents the facilities 1-4 respectively, numbers in parentheses ( ) above facilities are the processing times. Numbers in square brackets [ ] below or between facilities represent buffer capacities. For instance, facilities 2 and 3 can process a maximum of two jobs simultaneously. Considering these structures, the relevant representation matrices are set as follows:

$$P_k = \text{diag}[1 \ 2 \ 4 \ 3], \quad F = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}, \quad C = \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ e \end{bmatrix}^T, \quad Q = \begin{bmatrix} 1 & \varepsilon & \varepsilon & 6 \\ \varepsilon & 2 & 4 & \varepsilon \\ e & e & 2 & \varepsilon \\ \varepsilon & \varepsilon & e & 1 \end{bmatrix}$$

The transit time from facility 1 to 3 is positive and finite, and fluctuates periodically as:

$$\tau_{31}(k) = \{0.5, 1, 0.5, 1, \dots\}$$

which is dependent on the job number. Recalling sect. 4.6, install a new imaginary facility 5 between facilities 1 and 3, and modify the relevant representation matrices. The number of jobs to process is  $k = 16$ , and let all required materials be ready at time  $t = 0$ . This indicates  $u(k) = [0 \ 0]^T \ (1 \leq k \leq 16)$ . Moreover, assuming the initial condition is empty, yields  $x(0) = \varepsilon \in R_{\max}^{10 \times 1}$ .

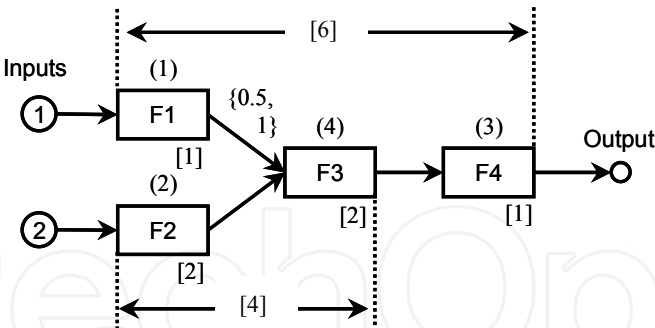


Fig. 6. A Simple manufacturing system

Fig. 7 shows the earliest process start time in facilities 1-4. The horizontal axis represents job number  $k$ . Looking at the system as a whole, the facility with the highest processing ability is 1, and the lowest is 4. In facility 1, the earliest time depends on its processing ability for  $1 \leq k \leq 6$ , but for  $k \geq 7$ , it comes to depend on the process completion times in facility 4 due to the capacity constraint between them. Facility 2 can process two jobs at maximum simultaneously, the facility processes jobs in accordance with its own ability in  $1 \leq k \leq 4$ . However, for  $k \geq 5$ , it is limited by the capacity constraint of facility 3. Facility 3 can also process two jobs at the same time, which implies that its effective throughput is greater than

facility 4. Thus, as the job number  $k$  grows and the system approaches a stationary state, the entire throughput becomes dependent on facility 4 which has the lowest processing ability.

Next, let us consider a system reschedule. Suppose facility 3 breaks down for a period during processing with  $k=3$ , delaying completion for 10 time units. The results for recalculating the schedule using (34) for  $k=3$ , and (22) for  $k \geq 4$  are shown in Fig. 8. The first effect of this change on the succeeding facility 4 for  $k \geq 3$ , followed by facility 2 that has its capacity constrained by facility 3 when  $k \geq 7$ . Moreover, facility 1's capacity is constrained by facility 4 when in  $k \geq 9$ . The relative values between facilities for  $k=10$  are similar to those for  $k=2$  in Fig. 7, which implies that the through-puts in facilities 1-3 may become subordinated to facility 4.

Let us now consider an example of on-line monitoring that uses both forward and backward state-space representations. Fig. 9 shows the float times in facilities 1-4 when the required output times are equal to those in Fig. 8. Float times are derived from the difference between the latest and earliest starting times; a negative value means that there is no float in the corresponding facility. Facility 4, which is located furthest downstream, is affected by the delay in  $k=3$  immediately. This delay affects facilities 1-3, upstream, only after several jobs have been processed. As the processing proceeds, facilities 1-3 regain float times, with facility 2 holding the largest as it can process two jobs simultaneously. Although the effective throughput of facility 1 is equal to facility 2, it can process only one job at a time. Thus, its float time is comparable to facility 3.

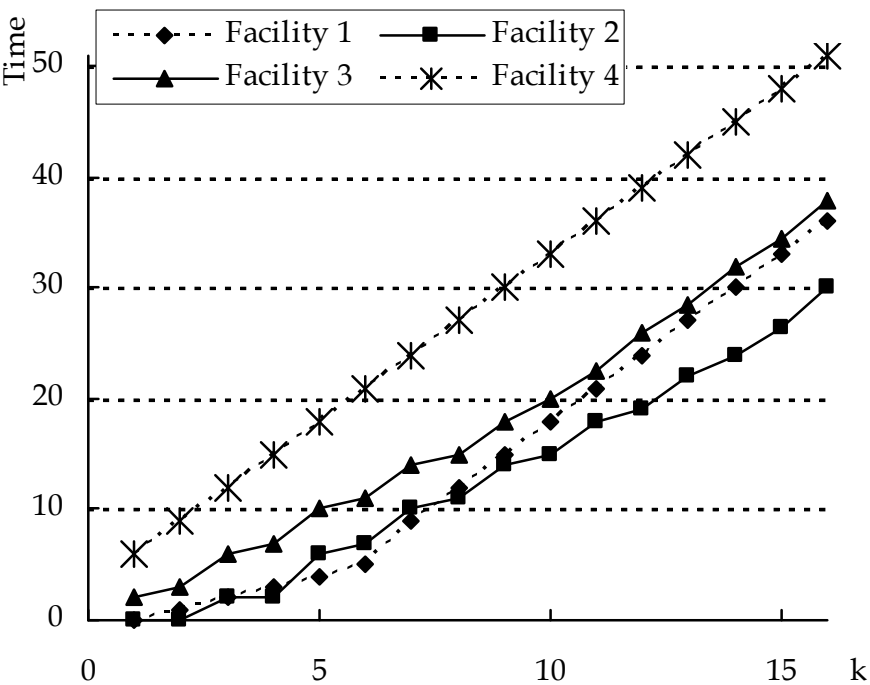


Fig. 7. Earliest starting times in facilities 1-4

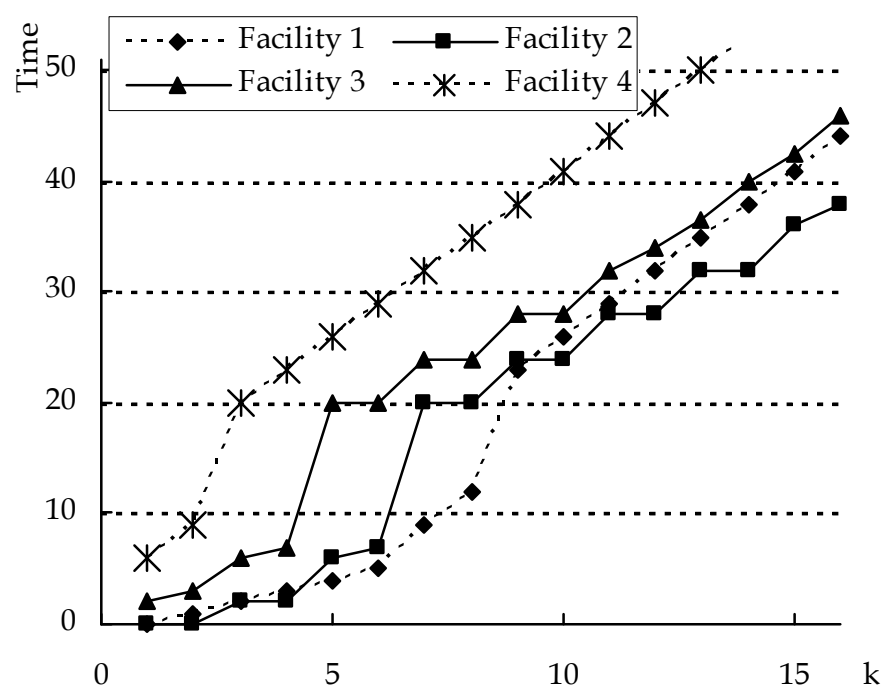


Fig. 8. Results of rescheduling

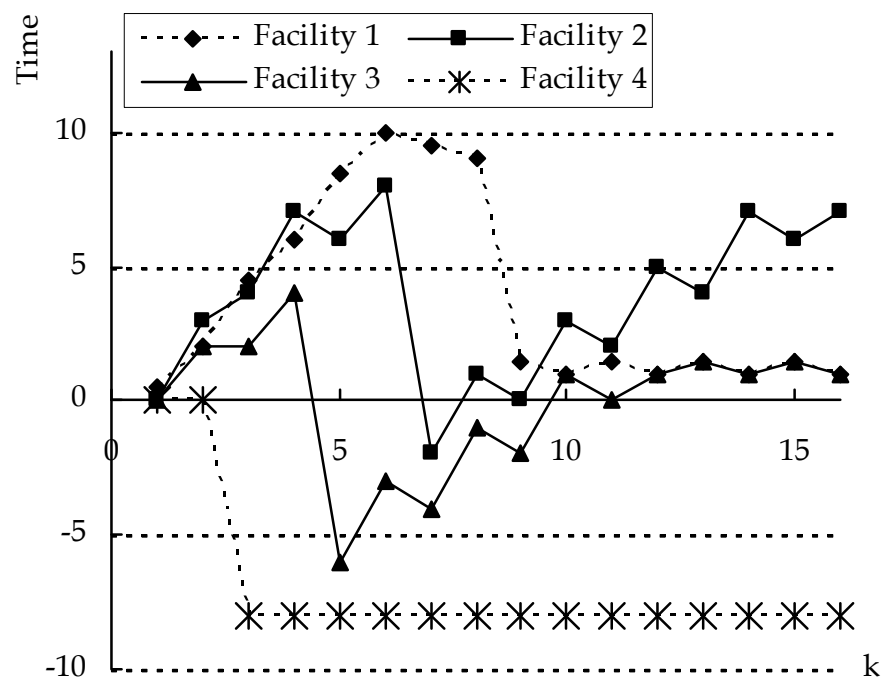


Fig. 9. Float times in facilities 1-4

7. Conclusion and future insights

This chapter has introduced modelling methods for a class of discrete event systems. Specifically, we have focused on and extended the state-space representation in Dioid and



max-plus algebras. The simplest representation can only describe the behaviour of systems in which the buffer capacities are one in single facilities, and infinite between two adjacent facilities. This constraint is restrictive when applying the representation to practical systems. To resolve this, we have intensively worked on developing a systematic framework to derive state-space representations for systems where the capacity constraints for a single or between two arbitrary facilities can be taken into account. Two types of representations called forward and backward, were derived, by which the earliest and latest process start and completion times can be calculated. By using both, the float times of internal facilities can be calculated. In addition, we considered a rescheduling method that can be used for cases where the process start or completion times, or processing times are changed after the corresponding job has commenced. Using the derived formula, we can accomplish an on-line scheduling where the internal parameters change frequently.

Finally, we mention insights that point to future directions in this research field. First, the state-space representation should be extended to be able to consider the set of engaged facilities. This research assumed that all jobs use all facilities. However, in several systems, railway systems for instance, the set of facilities engaged for a particular job may differ. Moreover, capacity constraints in single facilities and between facilities are usually invoked. In existing methods, one or other constraint is taken into account, but no method that considers both of these simultaneously has been proposed. Such developments would be very important for practical applications. Second, efficient computation methods for the state equation should be developed. In terms of computation time, the time for computing the state equation increases rapidly as the system's size increases. Thus, developing efficient algorithms is essential for on-line operations. These issues should be of primary concern in future work with the potential to offer greater scope in applications.

## 8. References

- Baccelli, F. & Hong, D. (2000). TCP is max-plus linear: and what tells us on its throughput, *Proceedings of ACM-SIGCOMM*, pp.219–230, Stockholm, August 2000
- Baccelli, F.; Cohen, G., Olsder, G.J. & Quadrat, J.P. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*, John Wiley & Sons, North River Press, ISBN: 047193609X, New York. <http://maxplus.org>
- Boimond, J.L. & Ferrier, J.L. (1996). Internal model control and max-algebra: controller design. *IEEE Transactions on Automatic Control*, Vol. AC-41, No. 3, 457–461, ISSN: 0018-9286
- Boom, T.; Heidergott, B. & De Shutter, B. (2007). Complexity reduction in MPC for stochastic max-plus linear discrete event systems by variability expansion. *Automatica*, Vol.43, No.6, 1058–1063, ISSN: 0005-1098
- Boom, T.; De. Schutter, B., Schullerus, G. & Krebs, V. (2003). Adaptive model predictive control using max-plus-linear input-output models, *Proceedings of the American Control Conference*, pp.933–938, Denver, June 2003
- Girault, C. & Valk, R. (2002). *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*, Springer, ISBN: 3540412174, Berlin
- Cofer, D.D. & Garg, V.K. (1996). Supervisory control of real-time discrete event systems using lattice theory. *IEEE Transactions on Automatic Control*, Vol.41, No.12, 1751–1778, ISSN: 0018-9286

- Goto, H. & Takahashi, H. (2009). Fast computation methods for the Kleene star in max-plus linear systems with a DAG structure. *IEICE Transactions on Fundamentals*, Vol.E92-A, No.11, 2794-2799, ISSN: 0916-8508
- Goto, H. (2007). Dual representation of event-varying max-plus linear systems. *International Journal of Computational Science*, Vol. 1, No.3, 225-242, ISSN: 1992-6669
- Goverde, R. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B*, Vol. 41, No.2, 179-201, ISSN: 0191-2615
- Heemels, W.P.M.H.; De Schutter B. & Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, Vol.37, No.7, 1085-1091, ISSN: 0005-1098
- Harrison, J. (2009). *Handbook of Practical Logic and Automated Reasoning*, Cambridge University Press, ISBN: 0521899575, New York
- Heidergott, B.; Olsder, G.J. & Woude, J. (2006). *Max Plus at Work: Modeling and Analysis of Synchronized Systems*, Princeton University Press, ISBN: 0691117632, New Jersey
- Heidergott, B. & De Vries, R. (2001). Towards a (max,+) control theory for public transportation networks. *Discrete Event Dynamical Systems: Theory and Applications*, Vol.11, No.4, 371-398, ISSN: 0924-6703
- Kelarev, A. (2003). *Graph Algebras and Automata*, Marcel Dekker, ISBN: 0824747089, New York
- Krivulin, N. (1996). Max-plus algebra models of queuing networks, *Proceedings of International Workshop on Discrete Event Systems*, pp.76-81, London, August 1996
- Moh, A.; Manier, M., Manier H. & Moudni, A. (2005). A max-plus algebra modeling for a public transport system. *Cybernetics and Systems*, vol. 36, 1-16, ISSN: 0196-9722
- Ramadge, P.J. & Wonham, W.M. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, Vol.25, No.1, 206-230 ISSN: 0363-0129
- Sampath, M.; Sengupta, R., Lafortune, S., Sinnamohideen K. & Teneketzis, D. (1996). Failure diagnosis using discrete event models, *IEEE Transactions on Control System Technology*, Vol.4, No.2, 105-124, ISSN: 1063-6536
- Schullerus, G.; Krebs, V., De Schutter, B. & Boom, T. (2006). Input signal design for identification of max-plus-linear-systems. *Automatica*, Vol.42, No.6, 937-943, ISSN: 0005-1098
- Schullerus, G. & Krebs, V. (2001). Diagnosis of batch processes based on parameter estimation of discrete event models, *Proceedings of the European Control Conference*, pp.1612-1617, Porto, September 2001
- De Schutter, B.; Boom, T. & Verdult, V. (2002). State space identification of max-plus-linear discrete event systems from input-output data, *Proceedings of the 41st IEEE Conference on Decision and Control*, pp.4024-4029, Las Vegas, December 2002
- De Schutter, B. & Boom, T. (2001). Model predictive control for max-plus-linear systems. *Automatica*, Vol.37, No.7, 1049-1056, ISSN: 0005-1098

IntechOpen

IntechOpen



## **Discrete Event Simulations**

Edited by Aitor Goti

ISBN 978-953-307-115-2

Hard cover, 330 pages

**Publisher** Sciyo

**Published online** 18, August, 2010

**Published in print edition** August, 2010

Considered by many authors as a technique for modelling stochastic, dynamic and discretely evolving systems, this technique has gained widespread acceptance among the practitioners who want to represent and improve complex systems. Since DES is a technique applied in incredibly different areas, this book reflects many different points of view about DES, thus, all authors describe how it is understood and applied within their context of work, providing an extensive understanding of what DES is. It can be said that the name of the book itself reflects the plurality that these points of view represent. The book embraces a number of topics covering theory, methods and applications to a wide range of sectors and problem areas that have been categorised into five groups. As well as the previously explained variety of points of view concerning DES, there is one additional thing to remark about this book: its richness when talking about actual data or actual data based analysis. When most academic areas are lacking application cases, roughly the half part of the chapters included in this book deal with actual problems or at least are based on actual data. Thus, the editor firmly believes that this book will be interesting for both beginners and practitioners in the area of DES.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hiroyuki Goto (2010). Modeling Methods Based on Discrete Algebraic Systems, Discrete Event Simulations, Aitor Goti (Ed.), ISBN: 978-953-307-115-2, InTech, Available from: <http://www.intechopen.com/books/discrete-event-simulations/modeling-methods-based-on-discrete-algebraic-systems>

**INTECH**  
open science | open minds

#### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

#### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen